

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC918 U.S. PTO
09/755308
01/02/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
in this Office.

出 願 年 月 日
Date of Application:

2000年 2月14日

出 願 番 号
Application Number:

特願2000-034610

出 願 人
Applicant(s):

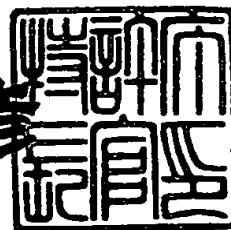
インターナショナル・ビジネス・マシーンズ・コーポレイシ
ョン

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 5月12日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3034165

【書類名】	特許願
【整理番号】	JP9200027
【あて先】	特許庁長官 殿
【国際特許分類】	G06F017/20
【発明者】	
【住所又は居所】	神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 大和事業所内
【氏名】	田所 瑞穂
【発明者】	
【住所又は居所】	神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 大和事業所内
【氏名】	乾 尚
【発明者】	
【住所又は居所】	神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 大和事業所内
【氏名】	月澤 一也
【特許出願人】	
【識別番号】	390009531
【住所又は居所】	アメリカ合衆国 1 0 5 0 4、ニューヨーク州アーモンク （番地なし）
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション
【代理人】	
【識別番号】	100086243
【弁理士】	
【氏名又は名称】	坂口 博
【選任した代理人】	
【識別番号】	100091568
【弁理士】	

【氏名又は名称】 市位 嘉宏

【手数料の表示】

【予納台帳番号】 024154

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 キャラクタ・ストリングを表示する方法

【特許請求の範囲】

【請求項 1】 コンピュータ・プログラムが提供する G U I 環境にキャラクタ・ストリングを表示する方法であって、

(a) 実行可能プログラムを提供するステップと、

(b) 前記キャラクタ・ストリングを含み前記実行可能プログラムから参照することが可能なテキスト・ファイルを提供するステップと、

(c) 前記実行可能プログラムを実行するステップと

(d) 前記実行可能プログラムから前記テキスト・ファイルを参照するステップと、

(e) 前記参照したテキスト・ファイルに含まれるキャラクタ・ストリングを表示するステップと

を有する表示方法。

【請求項 2】 コンピュータ・プログラムが提供する G U I 環境にキャラクタ・ストリングを表示する方法であって、

(a) 実行可能プログラムを提供するステップと、

(b) 前記キャラクタ・ストリングと検索キーを含み前記実行可能プログラムから参照することが可能なテキスト・ファイルを提供するステップと、

(c) 前記実行可能プログラムを実行するステップと

(d) 前記検索キーを検索するステップと、

(e) ステップ (d) で、前記検索キーがイネーブルであることを検出したときは前記テキスト・ファイルが含むキャラクタ・ストリングを参照するステップと、

(f) 前記参照したキャラクタ・ストリングを表示するステップと、

(g) ステップ (d) で、前記検索キーがディスエーブルであることを検出したときは前記実行可能プログラムに組み込まれたキャラクタ・ストリングを表示するステップと

を有する表示方法。

【請求項 3】 GUI 環境にキャラクタ・ストリングを表示することができるコンピュータ・プログラムの構造であって、

GUI 環境を定義するコードと、

前記キャラクタ・ストリングを含むテキスト・ファイルを参照するコードと、

前記参照したテキスト・ファイルに記述されたキャラクタ・ストリングを表示させるコードと

を含むコンピュータ・プログラムの構造。

【請求項 4】 GUI 環境にキャラクタ・ストリングを表示することができるコンピュータ・プログラムの構造であって、

GUI 環境を定義するコードと、

デフォルト・キャラクタ・ストリングと、

選択キャラクタ・ストリングを含むテキスト・ファイルの検索キーを検索するコードと、

前記検索キーがイネーブルであることを検出したとき、前記選択キャラクタ・ストリングを表示させるコードと、

前記検索キーがディスエーブルであることを検出したとき、前記デフォルト・キャラクタ・ストリングを表示させるコードと

を含むコンピュータ・プログラムの構造。

【請求項 5】 GUI 環境にキャラクタ・ストリングを表示することができるコンピュータ・プログラムの作成方法であって、

GUI 環境を提供するコードを記述するステップと、

前記キャラクタ・ストリングを含むテキスト・ファイルを参照するコードを記述するステップと、

前記参照したテキスト・ファイルに記述されたキャラクタ・ストリングを表示させるコードを記述するステップと

を含むコンピュータ・プログラムの作成方法。

【請求項 6】 GUI 環境にキャラクタ・ストリングを表示することができるコンピュータ・プログラムの作成方法であって、

GUI 環境を提供するコードを記述するステップと、

デフォルト・キャラクタ・ストリングを記述するステップと、
選択キャラクタ・ストリングを含むテキスト・ファイルの検索キーを検索するコードを記述するステップと、
前記検索キーがイネーブルであることを検出したとき、前記選択キャラクタ・ストリングを表示させるコードを記述するステップと、
前記テキスト・ファイルの検索キーがディスエーブルであることを検出したとき、前記デフォルト・キャラクタ・ストリングを表示させるステップとを含むコンピュータ・プログラムの作成方法。

【請求項7】 GUI環境にキャラクタ・ストリングを表示させることができるコンピュータ・プログラムを記録したコンピュータによる読み取り可能な記録媒体であって、

GUI環境を提供するコードと、
前記キャラクタ・ストリングを含むテキスト・ファイルを参照するコードと、
前記参照したテキスト・ファイルに記述されたキャラクタ・ストリングを表示させるコードと
を含むコンピュータによる読み取り可能な記録媒体。

【請求項8】 GUI環境にキャラクタ・ストリングを表示させることができるコンピュータ・プログラムを記録したコンピュータによる読み取り可能な記録媒体であって、

GUI環境を提供するコードと、
デフォルト・キャラクタ・ストリングを提供するコードと、
選択キャラクタ・ストリングを含むテキスト・ファイルの検索キーを検索するコードと、
前記検索キーがイネーブルであることを検出したとき、前記選択キャラクタ・ストリングを表示させるコードと、
前記検索キーがディスエーブルであることを検出したとき、前記デフォルト・キャラクタ・ストリングを表示させるコードと
を含むコンピュータによる読み取り可能な記録媒体。

【請求項9】 制御に関連するキャラクタ・ストリングを含むコンピュータ・

プログラムの作成方法であって、

(a) 前記キャラクタ・ストリングを記述したテキスト・ファイルを作成するステップと、

(b) 前記テキスト・ファイルを参照するコードを記述したプログラム・ファイルを作成するステップと

を含むコンピュータ・プログラムの作成方法。

【請求項 1 0】 制御に関連するキャラクタ・ストリングを含むコンピュータ・プログラムの作成方法であって、

(a) 選択キャラクタ・ストリングおよび検索キーを含むテキスト・ファイルを作成するステップと、

(b) デフォルト・キャラクタ・ストリングと、前記検索キーがイネーブルであることを検出したとき前記選択キャラクタ・ストリングに制御を移し前記検索キーがディスエーブルであることを検出したとき前記デフォルト・キャラクタ・ストリングに制御を移すコードとを含むプログラム・ファイルを作成するステップと

を含むコンピュータ・プログラムの作成方法。

【請求項 1 1】 コンピュータの制御に関連するキャラクタ・ストリングを含むコンピュータ・プログラムの構造であって、

前記キャラクタ・ストリングを含むテキスト・ファイルを参照するコードと、

前記参照したテキスト・ファイルに記述されたキャラクタ・ストリングを前記コンピュータの制御に利用するコードと

を含むコンピュータ・プログラムの構造。

【請求項 1 2】 コンピュータの制御に関連するキャラクタ・ストリングを含むコンピュータ・プログラムの構造であって、

デフォルト・キャラクタ・ストリングと、

選択キャラクタ・ストリングを含むテキスト・ファイルの検索キーを検索するコードと、

前記検索キーがイネーブルであることを検出したとき、前記選択キャラクタ・ストリングを前記コンピュータの制御に利用するコードと、

前記検索キーがディスエーブルであることを検出したとき、前記デフォルト・キャラクタ・ストリングを前記コンピュータの制御に利用するコードとを含むコンピュータ・プログラムの構造。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、修正の容易なコンピュータ・プログラムの作成方法等に関する。さらに、本発明はコンピュータ・プログラムが提供するGUI環境に表示するキャラクタ・ストリングやコンピュータの動作に関連するキャラクタ・ストリングを修正し易いプログラムの作成方法等に関する。さらに、本発明は多国言語をサポートし易いプログラムの作成方法等に関する。

【0002】

【従来の技術】

ユーザの特定の利用目的に向けて開発されるアプリケーション・プログラム（以後アプリケーションという。）は、ユーザに様々なメッセージを提供してユーザと対話しながら実行される。パーソナル・コンピュータにおいてはGUI（グラフィック・ユーザ・インターフェース）というヒューマン・インターフェースを採用し、コンピュータに対する処理の実行指令（コマンド）や機能を表すアイコン、文字、および図形などを使って、直観的にそれらの意味や機能を理解できるようにしている。

【0003】

Windows（マイクロソフト社の商標）やOS/2（IBM社の商標）などのGUI環境を提供するOS（オペレーティング・システム）のもとで動作するアプリケーションは、OSのライブラリに含まれるAPI（アプリケーション・プログラム・インターフェース）を利用してユーザに対するコマンドやデータの入力環境、入力方法、および結果等をウインドウ上に文字や図形で表示できる。アプリケーションの世界的な流通の拡大に伴い、ある国の言語により開発されたアプリケーションであっても、他の言語を使用する国で利用する機会が急速に増えてきており、文字を多くの国の言語に対応させるニーズが高まってきている

【 0 0 0 4 】

通常アプリケーションはそれが動作するOS環境の言語と同一になるように作成する。例えば英語のOSは、そのGUIにおけるメッセージを英語で表示しており、したがってそのOS上で動作するアプリケーションはメッセージを英語で表示するように開発する。さらに、アプリケーションがOSをコールするときは、アプリケーションからOSにキャラクタ・ストリングのメッセージを送り、OSはキャラクタ・ストリングの文字列、英語、日本語等の言語の種類、全角、半角等の文字形式等を判断して、所定の定義に合致した場合のみ正しいメッセージとして受け取り後の処理を実行する場合もある。

【 0 0 0 5 】

いま、英語のOS環境で開発された英語のアプリケーションをフランス人が利用する場合、アプリケーションのキャラクタ・ストリングはフランス語に翻訳することが好ましい。これはフランス人にとってたとえOSは英語であったとしてもアプリケーションのGUIに表示されるメッセージはフランス語のほうが使いやすいからである。

【 0 0 0 6 】

アプリケーションは、通常、CPUが実行する命令を記述したコード部分とそのパラメータとしてのデータ部分の両方を含む。アプリケーションが含んでいるデータのひとつのタイプをリソースと称する。リソースには、アイコン、カーソル、メニュー、ダイアログ・ボックス、および特定の言語の文字等を含んでいる。リソースはGUIおよびマルチメディア・プログラム等で広範囲に使用される。

【 0 0 0 7 】

いま、多くの国の言語に対応できるアプリケーションを開発するとき、メイン・ファイルとそのサブルーチンすなわちリソース・ファイルとに分割して作成すると都合がよい。アプリケーションはアセンブラ、COBOL、FORTRAN、C/C++、PL1等の人間が理解できるプログラム言語によってソース・コード形式で記述される。

【0008】

リソース・ファイルには、GUIやOSに提供するメッセージとして特定の言語および形式で作成されたキャラクタ・ストリングが含まれる。ある国の言語に基づいてソース・プログラムを作成した後にそれを他の国向けに書き換えるときは、メイン・ファイルのソース・コードは変更しないで、リソース・ファイルのソース・コードのメッセージに相当するキャラクタ・ストリングを翻訳して行う。ソース・コードはコンピュータが理解できないので、つぎに両者をそれぞれコンパイルしてオブジェクト・コードに変換し、適宜OSが提供するライブラリを組み込んで実行可能プログラムを作成する。オブジェクト・コードは機械語ともいい「0」と「1」で表現される。

【0009】

メイン・ファイルは実行の過程でリソース・ファイルを参照し、リソース・ファイルに含まれたメッセージを示すキャラクタ・ストリングをGUIを通じてユーザに提供したり、直接OSに提供する。したがってアプリケーションのメッセージを他の国の言語に翻訳して動作させようとする場合には、ソース・コードで記述されたリソース・ファイルのなかのキャラクタ・ストリングを翻訳して、各国ごとのリソース・ファイルを用意し、再度コンパイルして実行可能プログラムを作成する必要がある。

【0010】

特開平10-116185号公報には、このようなリソース・ファイルを作成する技術が開示されている。また、米国特許5644775号公報には、アプリケーションのテキスト部分をプログラム本体から分割してライブラリに登録し、メッセージを多くの国の言語に翻訳することを容易にする技術が開示されている。ライブラリは複数のアプリケーションに関連したテキストを表示するキャラクタ・ストリングと、特定の1つのアプリケーションに関連したキャラクタ・ストリングを含む。

【0011】

【発明が解決しようとする課題】

アプリケーションが多数の国の言語をサポートしようとするとき、従来の方法

ではソース・コードで記述されたリソース・ファイルのメッセージに相当するキャラクタ・ストリングを翻訳したのち、リソース・ファイルをコンパイルしてメイン・ファイルとともに実行可能プログラムを作成し、その動作を確認する必要があった。開発元では、あらかじめソース・コードで記述したアプリケーションをサポートする言語の数だけ作成し、それぞれコンパイルしてオブジェクト・コードからなる実行可能プログラムに変換して出荷していた。したがって、一旦出荷した後に、サポートしていない言語の必要性が出現した場合は、開発元においてソース・コードの書き換えを行うしかなかった。

【 0 0 1 2 】

また、日本で、日本語フォントをサポートする英語のOS環境下で英語のアプリケーション・プログラムを動作させる場合には、重要なメッセージだけでも日本語で表示したいという要望もあった。たとえば、ポータブル・コンピュータのバッテリー残量を測定し、ロー・レベルに達したときにはACアダプタを挿入するようにユーザに促すメッセージは、作業データの喪失を防止する観点から重要であり、翻訳することがプログラムの動作に影響がない限りこの部分だけでも日本語にしておきたい場合があった。

【 0 0 1 3 】

さらにまた、OSの言語環境が同一であっても、そのバージョンがアップしたときに、それまでのアプリケーションが動作しないという問題もあった。リソース・ファイルのキャラクタ・ストリングはOSに対するメッセージとして利用される。日本語OSの環境下で開発されたアプリケーションでも、OSのバージョンがアップしたときに、当初は全角のメッセージだけをOSが受付ており、バージョン・アップ後は半角のメッセージだけを受け付けるように変更されている場合がある。このときは、ソース・コードで記述されたリソース・ファイルのキャラクタ・ストリングを半角で記述し直す必要がある。これらの種々の問題から、メイン・ファイルやリソース・ファイルのソース・コードを修正しないでキャラクタ・ストリングを簡易に変更できる手法が待望されていたのである。

【 0 0 1 4 】

したがって、本発明の目的は修正の容易なコンピュータ・プログラムの作成方

法、プログラム構造、および記憶媒体を提供することになる。さらにまた、本発明の目的は、ソース・プログラムを修正しないで、キャラクタ・ストリングを修正できるコンピュータ・プログラムの作成方法、プログラム構造、および記憶媒体を提供することにある。

【 0 0 1 5 】

本発明のさらに詳細な目的は、キャラクタ・ストリングの表示を含む G U I 環境を提供するコンピュータ・プログラムにおいて、言語環境の変更またはプログラムの動作環境の変化に対応するために、キャラクタ・ストリングの修正を容易に行えるコンピュータ・プログラムの作成方法、プログラム構造、記憶媒体を提供することにある。

【 0 0 1 6 】

【課題を解決するための手段】

本発明の原理は、G U I に提供するメッセージや、プログラムの動作に関連して他のプログラムと交換するメッセージに相当するキャラクタ・ストリングを含むコンピュータ・プログラムの当該キャラクタ・ストリングを、テキスト・ファイルとして独立して作成したことにある。テキスト・ファイルはプログラムから参照されてキャラクタ・ストリングをプログラムに提供する。テキスト・ファイルはプログラムから参照される形式を備えているためメッセージを修正してもプログラム自体を変更する必要はない。テキスト・ファイルの変更はテキスト・エディタを利用して極めて容易に行える。

【 0 0 1 7 】

本発明の一の側面は、コンピュータ・プログラムが提供する G U I 環境にキャラクタ・ストリングを表示する方法であって、(a) 実行可能プログラムを提供するステップと、(b) 前記キャラクタ・ストリングを含み前記実行可能プログラムから参照することが可能なテキスト・ファイルを提供するステップと、(c) 前記実行可能プログラムを実行するステップと (d) 前記実行可能プログラムから前記テキスト・ファイルを参照するステップと、(e) 前記参照したテキスト・ファイルに含まれるキャラクタ・ストリングを表示するステップとを有する。

【 0 0 1 8 】

テキスト・ファイルはキャラクタ・ストリングを含み、実行可能プログラムから参照できるように構成されている。したがって、実行可能プログラムはテキスト・ファイルのキャラクタ・ストリングを参照して G U I に表示することができ、表示すべきキャラクタ・ストリングを変更するときは、変更したテキスト・ファイルを提供するだけでよい。ここに、キャラクタ・ストリングは G U I 環境に表示すると同時に、他のプログラムのメッセージとしても利用されるものであってもよい。また、本発明のプログラムは、アプリケーションに限定されるものではなく、他のサービス・プログラム、デバイス・ドライバ等のプログラムであっても、G U I のメッセージや他のプログラムとの間で交換するメッセージに相当するキャラクタ・ストリングを含むプログラムであればよい。

【 0 0 1 9 】

本発明の他の側面は、コンピュータ・プログラムが提供する G U I 環境にキャラクタ・ストリングを表示する方法であって、(a) 実行可能プログラムを提供するステップと、(b) 前記キャラクタ・ストリングと検索キーを含み前記実行可能プログラムから参照することが可能なテキスト・ファイルを提供するステップと、(c) 前記実行可能プログラムを実行するステップと (d) 前記検索キーを検索するステップと、(e) ステップ (d) で、前記検索キーがイネーブルであることを検出したときは前記テキスト・ファイルが含むキャラクタ・ストリングを参照するステップと、(f) 前記参照したキャラクタ・ストリングを表示するステップと、(g) ステップ (d) で、前記検索キーがディスエーブルであることを検出したときは前記実行可能プログラムに組み込まれたキャラクタ・ストリングを表示するステップとを有する。

【 0 0 2 0 】

この構成を備える結果、テキスト・ファイルの検索キーがイネーブルのときは、テキスト・ファイルのキャラクタ・ストリングを参照し、ディスエーブルのときは実行可能プログラムに当初から含んでいたキャラクタ・ストリングを利用する。よって、実行可能プログラムの G U I 環境の言語を翻訳したいときや、O S の動作環境に変更があった場合のみテキスト・ファイルを作成して対応できる。

いいかえると、プログラム中のキャラクタ・ストリングをテキスト・ファイルを編集するだけで制御できるということである。

【0021】

本発明の他の側面は、このようなコンピュータ・プログラムの構造、作成方法、記録媒体に向けられる。さらにまた、本発明の他の側面は、制御に関連するキャラクタ・ストリングを含むコンピュータ・プログラムの作成方法に関する。この側面においては、当該キャラクタ・ストリングをテキスト・ファイルによりプログラムに提供することができるため、テキスト・ファイルの書き換えだけでキャラクタ・ストリングがOS環境と一致しない不都合を容易に解消できる。

【0022】

【発明の実施の形態】

[コンピュータの構成]

図1は本発明に係るアプリケーションを実行するコンピュータの概略ブロック図である。CPU（中央演算処理装置）101は、コンピュータ100の心臓部に当たり、コンピュータの入出力、命令の実行などすべての処理を行う。メイン・メモリ103はプログラムやデータを一時的に記憶する装置で、コンピュータ内部でCPU101がプログラムの実行やデータ処理を行う際、当該プログラムをCPUに転送するために記憶したり作業結果を記憶したりする領域を提供する。HDD105は、OS、BIOS、デバイス・ドライバ、アプリケーション・プログラム等のコンピュータ100で実行するための各種プログラムやデータを格納しておき、必要に応じてメイン・メモリ103や他のコンポーネントに転送される。メイン・メモリ103に記憶されたプログラムやデータは、コンピュータの電源を停止すると消失するが、HDD105に格納されたプログラムやデータは電源を停止しても維持することができる。

【0023】

ディスプレイ107は、キーボード113やマウス115と共にヒューマン・インターフェースを実現するための装置で、GUIの環境下ではユーザにグラフィカルな出力情報を提供したり、コンピュータの操作やプログラムの実行のための入力環境を提供する。キーボード113はユーザが数字、文字、記号等でコン

コンピュータに指示を与える機能を提供する。マウス 1 1 5 は、G U I 環境のもとでディスプレイに表示されたカーソルを移動し、ディスプレイ上のアイコンにアクセスしてコマンドを実行したり、プログラムに各種指示を与える働きをする。F D D (フロッピー・ディスク・ドライブ) 1 0 9 はフロッピー・ディスク駆動用の装置で、C D - R O M ドライブ 1 1 1 は C D - R O M の読み取り専用装置である。通信コネクタ 1 1 7 は、インターネットやイントラネット等のネット・ワークをコンピュータに接続するためのインターフェースである。

【 0 0 2 4 】

H D D 1 0 5 には、G U I 環境をサポートする W i n d o w s や O S / 2 などの O S が格納されており、起動時にメモリに読み出される。O S はアプリケーションが G U I を実現するために必要な基本的な機能をライブラリに格納した A P I として提供する。ライブラリにはコンパイル時に実行ファイルに組み込まれるスタティック・リンク・ライブラリや、実行ファイルが動的に参照するダイナミック・リンク・ライブラリ (D L L) などが含まれる。

【 0 0 2 5 】

図 1 は本発明の実施例を説明する上で必要な範囲で概略的に記載したものであり、実際のコンピュータにはさらに多くのコンポーネントが搭載され、その接続関係も異なったもので実現されることは当業者にとって明らかである。したがって、本発明の趣旨を逸脱しない範囲でこれと異なるいかなる態様のコンピュータにおいても本発明のプログラムは実行できる。

【 0 0 2 6 】

図 2 に本発明の実施例に係るアプリケーションの作成方法を描いたフローチャートを示す。図 3 には、アプリケーションがディスプレイ上で提供する G U I に相当するウインドウを示す。いま、コンピュータ 1 0 0 で動作する O S は日本語、英語およびフランス語のフォントをサポートしており、アプリケーションは英語で作成されており、ウインドウのキャラクタ・ストリングだけ日本語またはフランス語に翻訳したい場合を考える。図 3 は、本実施例においてアプリケーションが提供するウインドウ内におけるキャラクタ・ストリングがリリース時は英語 (A) で表示され、その後日本語 (B) またはフランス語 (C) に翻訳して表示

する場合のウインドウのイメージを示している。

【 0 0 2 7 】

本実施例のアプリケーションはソース・コードで記述したメイン・ファイル（ブロック 2 0 1）、ソース・コードで記述したリソース・ファイル（ブロック 2 0 3）からなる。しかし、本発明の趣旨は、アプリケーションがいかなるモジュール単位に分割されているかを問題にしないことは、後の説明で明らかになるであろう。また、これらのコードは任意のプログラム言語で記述することができる。

【 0 0 2 8 】

[メイン・ファイルの構造]

メイン・ファイルは当該プログラムの主たる機能を規定する部分であり、一般的に OS の言語環境に合致したキャラクタ・ストリングをウインドウに提供するように作成されている。メイン・ファイルのソース・コードの一部を図 4 に示す。メイン・ファイルを実行する過程において、ウインドウ上にキャラクタ・ストリングのメッセージを提供する場合にはメイン・ファイルからリソース・ファイル（ソース・コードは図示せず。）をコールしてリソース・ファイルに用意されたメッセージを表示するようになっている。また、メイン・ファイルは実行時にテキスト・ファイル（後に説明する。）中の検索キーの状態を検索するコードを含む。検索キーはアプリケーションがウインドウに提供すべきメッセージを示すキャラクタ・ストリングが、リソース・ファイルではなくてテキスト・ファイル中で定義されていることを示す。

【 0 0 2 9 】

メイン・ファイルを実行して、検索キーがイネーブルであることを発見したときは、テキスト・ファイル中のキャラクタ・ストリングを表示できるようになっている。検索キーがディスエーブルであることを発見したときは、リソース・ファイル中にデフォルトとして記述されたキャラクタ・ストリングを表示できるようになっている。すなわち、メイン・ファイルはテキスト・ファイルのキャラクタ・ストリングを表示するように検索キーが設定されていればテキスト・ファイルのキャラクタ・ストリングを表示し、検索キーが設定されていなければリソ

ース・ファイルのメッセージをデフォルトとして表示することができる。

【 0 0 3 0 】

[リソース・ファイルの構造]

リソース・ファイルはメイン・ファイルに関するグラフィックス・オブジェクトを定義する部分で、アイコン、カーソル、メニュー、ダイアログ・ボックス、および特定の言語の文字等のデータからなり、ディスプレイ上で当該プログラム専用のヒューマン・インターフェースを提供する。リソース・ファイルは主としてユーザにメッセージを提供する文字を他くの国の言語に翻訳する際の利便性を考慮して考え出されたファイルである。

【 0 0 3 1 】

リソース・ファイルは、プログラム言語で記述され、命令に相当するコード部分とメッセージに相当するキャラクタ・ストリング部分を含む。従来、メッセージを示すキャラクタ・ストリングを翻訳するときは、リソース・ファイルのソース・コードのうちキャラクタ・ストリングに関する部分だけを翻訳して実現することができたが、本発明ではリソース・ファイルを変更する必要はない。リソース・ファイルはメイン・ファイルからの指示を受けてデフォルトのキャラクタ・ストリングをディスプレイに表示する。

【 0 0 3 2 】

[実行可能ファイル]

つぎに、メイン・ファイルおよびリソース・ファイルのソース・コードをコンパイルして実行可能プログラム 2 1 1 を作成する。本実施例に係るアプリケーションは、GUI をサポートする特定の OS 環境下で開発されている。メイン・ファイルおよびリソース・ファイルはアプリケーションの本質的な部分だけを記述している。したがって、この二つのファイルのオブジェクト・コードだけでは GUI を提供しながら特定の機能を発揮するように、コンピュータ上で動作することができず、OS が提供するサービスを利用する必要がある。

【 0 0 3 3 】

したがって、メイン・ファイルおよびリソース・ファイルのソース・コードをそれぞれコンパイルしてオブジェクト・コードを作成した後は、連係編集プログ

ラム（リンク）を利用して、それぞれのオブジェクト・コードを結合したり、OSのライブラリ（ブロック205）に含まれているスタティック・リンク・ライブラリをオブジェクト・コードに組み込んだり、ダイナミック・リンク・ライブラリとリンクが可能なようにオブジェクト・コードに関連づけを行う。このようにして開発された実行可能プログラムは、開発時のOS環境と同一の環境下に導入されたとき動作できるようになる。

【0034】

ここまでの段階で特徴的なことは、このアプリケーションの実行可能プログラムはテキスト・ファイルを利用するか否かにかかわらず、デフォルトとしてリソース・ファイルに記述したキャラクタ・ストリングを利用して実行できるということである。開発段階が終了すると、アプリケーション・プログラムはFDDやCD-ROMの記録媒体に記録されてパッケージとして配布されたり、コンピュータ100の通信コネクタ117を利用してネットワーク経由で導入されたりする。

【0035】

[テキスト・ファイルの構成]

本発明の特徴として、キャラクタ・ストリングをプログラムの利用段階でも容易に記述できることが挙げられる。実行可能ファイルは、ウインドウへのメッセージの表示も含めて当初のOS環境のもとで完全に動作するように作成されている。しかし、リリースした後にウインドウのメッセージ部分だけを他の国の言語に翻訳する必要性が生ずる場合がある。また、リソース・ファイルに記述されているキャラクタ・ストリングは、アプリケーションの実行時にOSに対するメッセージとして送られる。OSがアプリケーションの開発段階からバージョン・アップした場合に、たとえOSの言語は同一であったとしても、全角と半角の違いなどでバージョン・アップ後のOS環境ではアプリケーションが正常に動作しないため、リソース・ファイルのキャラクタ・ストリングとは異なるキャラクタ・ストリングをメイン・ファイルに参照させるようにしたい場合がある。

【0036】

テキスト・ファイルはブロック219で提供される。ブロック219は開発段

階でなく、リリース後の利用段階であることは重要な意義を有する。テキスト・ファイルはプログラム・コードとは異なり、アセンブラー、コンパイラー、およびインタープリターを使ってオブジェクト・コードに変換してコンピュータに実行させるものではなく、GUIへの表示またはプログラム間のメッセージとして使用するためのキャラクタ・ストリングを提供するためにテキスト・エディタで編集できるファイルである。テキスト・ファイルはOSの環境が許容すれば、ANSI(米国規格協会)が制定したASCII(American Standard Code for Information Interchange)、JIS(日本工業規格)で規格化されたJISコード(JIS X 0208)、シフトJISやEUC(extended unix code)などのいかなる文字コードセットを利用してもよい。OSがWindowsならばマイクロ・ソフト社により規定されたINI形式のファイルを使用することができる。INI形式のファイルは検索に優れているからである。

【 0 0 3 7 】

図5は日本語(A)、フランス語(B)の2種類の言語に対応したテキスト・ファイルの実施例を示す。本発明の実施例において提供するテキスト・ファイル219は、実行可能プログラムから検索できる検索キーを含んでいる。図5において、“LD”が検索キーで、“LD=YES”は検索キーがイネーブルであることを示す。図4に示すメイン・ファイルのBOOL CSampleDlg::OnInitDialogを実行すると、GetINIString関数を呼び出し、GetINIString関数はテキスト・ファイルの検索キーを検索してイネーブルであることを検出すると、テキスト・ファイルで“IDS_STR_...”と表示された各識別子がついたキャラクタ・ストリングをテキスト・ファイルから読み出してくる。

【 0 0 3 8 】

いま、図5(A)のテキスト・ファイルの識別子をイネーブルにしておけば、アプリケーションの実行により図3(B)の日本語のキャラクタ・ストリングを含むウィンドウが提供される。図5の(B)のテキスト・ファイルの識別子をイネーブルにしておけば、図3(C)に示すフランス語のキャラクタ・ストリングを含むウィンドウが提供される。テキスト・ファイル219はプログラムの販売またはサービスに係わる技術員等(以後プログラムの最終使用者も含めてユーザ

という。)が作成することができる。

【0039】

[セット・アップ]

作成されたテキスト・ファイルは実行可能プログラムから検索できるようにHDD105内の共通のディレクトリ領域内に格納する。さらに、好ましくは、実行可能プログラム211のパッケージの中にディレクトリの関連付けを行った何も記述していないテキスト・ファイルを用意しておく。実行可能プログラムのパッケージには通常コンピュータへのプログラムの導入を補助するセット・アップ・プログラム(ブロック217)を含んでいる。

【0040】

セット・アップ・プログラムを使ってプログラムの導入をする際、セット・アップ・プログラムからメッセージ部分を他の国の言語へ翻訳する必要があるかどうか、バージョン・アップしたOS対応のために変更する必要があるかどうかの問い合わせをし、ユーザはその指示にしたがって、翻訳または変更したメッセージをテキスト・ファイルに書き込んでいく。この結果、ユーザはテキスト・ファイルのディレクトリを選択する作業や、不必要な部分まで翻訳してプログラムの動作に影響を与えるトラブルから解放される。

【0041】

ここまでの説明で明らかになったことは、アプリケーションのGUIに対するメッセージや動作に関連するキャラクタ・ストリングを翻訳、変更する作業が、開発者のもとで行われるソース・コードの修正を伴わないことである。テキスト・ファイルの編集者は、検索キーと識別子とキャラクタ・ストリングを記述するのみで実行可能ファイルもソース・プログラムも何ら変更する必要がない。

【0042】

[HDDへの格納]

実行可能プログラム215をコンピュータで動作できるようにするために、セット・アップ・プログラムはアプリケーションをHDD105に格納する(ブロック221)際に、格納場所の確定、起動条件の設定、作業ファイルがある場合はそのメモリ領域の確定、リンクするライブラリがある場合はリンク条件の設定

などを行う。以上までの手順により、アプリケーションをコンピュータ 1 0 0 において実行する準備が完了した。

【 0 0 4 3 】

図 6 は、アプリケーションを実行するときの手順を記述したフローチャートの実施例である。G U I 環境では、プログラムの起動を示すコマンドはアイコン形式でディスプレイ 1 0 7 上に提供され、ユーザは、マウス 1 1 5 を操作して起動コマンドを実行する（ブロック 3 0 1）。プログラムが起動されると実行可能プログラムはメイン・メモリに読み込まれる（ブロック 3 0 3）。このとき実行可能プログラムの一部だけメイン・メモリに読み込んでもよい。たとえば、リソース・ファイル部分は H D D に残しておき、メイン・ファイル部分からのコールがあったときだけメイン・メモリに必要な部分だけを転送し実行されるようにしてもよい。ダイナミック・リンク・ライブラリは実行可能プログラムからコールされたときだけメイン・メモリに読み込まれて実行される。

【 0 0 4 4 】

実行可能プログラムのメイン・ファイル部分は適宜リソース・ファイル部分を呼び出して G U I 環境を提供しながら、所定の機能を実行していく。リソース・ファイル部分は主としてディスプレイの表示に係わる部分を定義している。さらにメイン・ファイル部分はテキスト・ファイルに記述された検索キーを検索する（ブロック 3 0 7）。検索キーは、リソース・ファイルに当初から埋め込んであるキャラクタ・ストリングを利用するか、テキスト・ファイルに記述したキャラクタ・ストリングで置き換えるかを制御する。

【 0 0 4 5 】

ブロック 3 0 7 で検索キーがイネーブルのときは、当該テキスト・ファイルのキャラクタ・ストリングをメイン・メモリに読み出して、ディスプレイ 1 0 7 に表示したり、O S へメッセージとして送る。ブロック 3 0 9 で検索キーがディスエーブルのときは、メイン・ファイルはリソース・ファイルのキャラクタ・ストリングをメイン・メモリに読み出してディスプレイ 1 0 7 に表示したり、O S へメッセージとして送る。テキスト・ファイルから読み出したキャラクタ・ストリングは、メイン・ファイルからみたときリソース・ファイルのキャラクタ・スト

リングを置き換えたのと同様の機能を発揮する。したがって、テキスト・ファイルのキャラクタ・ストリングは、単にメッセージとして翻訳された言語で表示されるという側面だけでなく、メイン・ファイルとOSとの間でメッセージ交換するときのキャラクタ・ストリングの形式上の整合を図って動作上の支障を生じさせないという側面も兼ね備える。

【0046】

本実施例では、アプリケーションをメイン・ファイルとリソース・ファイルとに分けているが、GUIの定義もすべてメイン・ファイル内に記述してリソース・ファイルを使用しない場合でも同様の手法でテキスト・ファイルを作成することができる。また、テキスト・ファイルはテキスト・エディタでキャラクタ・ストリングを表示できるすべてのファイルを含む。

【0047】

【発明の効果】

本発明により、リリース後に修正の容易なコンピュータ・プログラムの作成方法、プログラム構造、および記憶媒体を提供することができた。さらにまた、本発明により、ソース・プログラムを修正しないで、GUIのメッセージや動作に関連するキャラクタ・ストリングを修正できるコンピュータ・プログラムの作成方法、プログラム構造、および記憶媒体を提供することができた。

【0048】

さらに本発明により、キャラクタ・ストリングの表示を含むGUI環境を提供するコンピュータ・プログラムにおいて、言語環境の変更またはプログラムの動作環境の変化に対応するために、キャラクタ・ストリングの修正を容易に行えるコンピュータ・プログラムの作成方法、プログラム構造、記憶媒体を提供できた。このことは、プログラムをリリースした後にアプリケーションのGUIがサポートしなければならない言語が増加した場合であっても、翻訳したメッセージを含むソース・コードを追加しなくてもよいという面でも有益である。

【図面の簡単な説明】

【図1】本発明の実施例に係るアプリケーションを実行するコンピュータの概略ブロック図である。

【図 2】本発明の実施例に係るアプリケーションの作成方法を示すフローチャートである。

【図 3】本発明の実施例に係るアプリケーションがディスプレイに提供する G U I を示す図である。

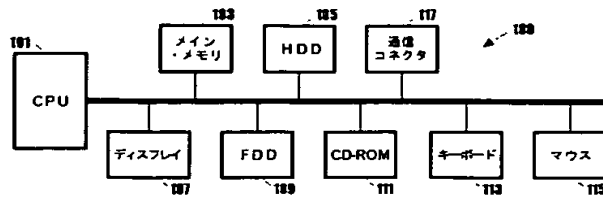
【図 4】本発明の実施例に係るアプリケーションのソース・コードを示す図である。

【図 5】本発明の実施例に係るテキスト・ファイルを示す図である。

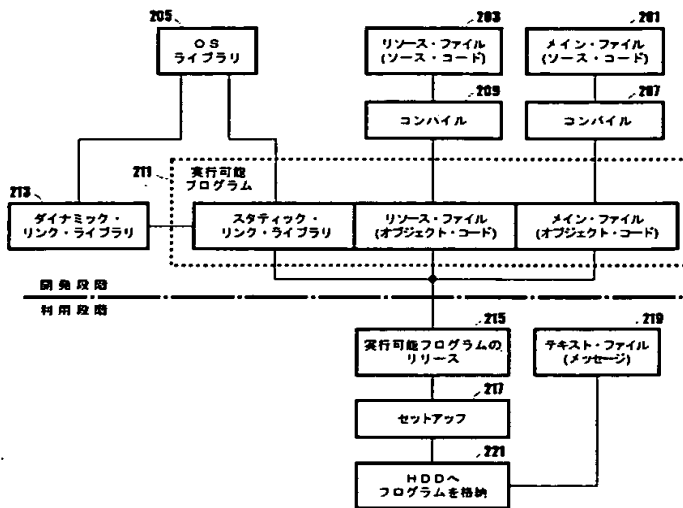
【図 6】本発明の実施例に係るアプリケーションを実行するときの手順を示したフローチャートである。

【書類名】 図面

【図 1】



【図 2】



【図 3】

☐ **Device Configuration** ✕

Port série	Enable
Parallelanschluß	Enable
Modem	Disable
Infrated	Disable
Audio	Enable

(A)

☐ **デバイス構成** ✕

シリアル・ポート	使用する
パラレル・ポート	使用する
モデム	使用しない
赤外線 ポート	使用しない
オーディオ	使用する

(B)

☐ **Configuration de périphérique** ✕

Port série	Activer
Port parallèle	Activer
Modem interne	Désactiver
Port infrarouge	Désactiver
Audio	Activer

(C)

【図 4】

```

// CSampleDlg message handlers
BOOL CSampleDlg::OnInitDialog()
{
    TCHAR szRet[128];

    CDialog::OnInitDialog();

    SetIcon(m_hIcon, TRUE);      // Set big icon
    SetIcon(m_hIcon, FALSE);    // Set small icon

    if (GetDlgItemText(IDS_STR_TITLE, T("IDS_STR_TITLE")))
        SetWindowText(szRet);
    if (GetDlgItemText(IDS_STR_SERIAL, T("IDS_STR_SERIAL")))
        SetDlgItemText(IDC_STATIC_SERIAL, szRet);
    if (GetDlgItemText(IDS_STR_PARA, T("IDS_STR_PARA")))
        SetDlgItemText(IDC_STATIC_PARA, szRet);
    if (GetDlgItemText(IDS_STR_MODEM, T("IDS_STR_MODEM")))
        SetDlgItemText(IDC_STATIC_MODEM, szRet);
    if (GetDlgItemText(IDS_STR_IR, T("IDS_STR_IR")))
        SetDlgItemText(IDC_STATIC_IR, szRet);
    if (GetDlgItemText(IDS_STR_AUDIO, T("IDS_STR_AUDIO")))
        SetDlgItemText(IDC_STATIC_AUDIO, szRet);
    if (GetDlgItemText(IDS_STR_OK, T("IDS_STR_OK")))
        SetDlgItemText(IDOK, szRet);
    if (GetDlgItemText(IDS_STR_HELP, T("IDS_STR_HELP")))
        SetDlgItemText(IDHELP, szRet);

    if (GetDlgItemText(IDS_STR_EN, T("IDS_STR_EN"))) {
        SetDlgItemText(IDC_STATIC_SERIAL1, szRet);
        SetDlgItemText(IDC_STATIC_PARA1, szRet);
        SetDlgItemText(IDC_STATIC_AUDIO1, szRet);
    }
    if (GetDlgItemText(IDS_STR_DIS, T("IDS_STR_DIS"))) {
        SetDlgItemText(IDC_STATIC_MODEM1, szRet);
        SetDlgItemText(IDC_STATIC_TR1, szRet);
    }

    // TODO: Add extra initialization here

    return TRUE; // return TRUE unless you set the focus to a control
}

//Load String
BOOL GetDlgItemText(TCHAR *pszRet, UINT nID, TCHAR *pszKey)
{
    TCHAR szIniFile[MAX_PATH], szRet[128];
    wsprintf(LPSTR, szIniFile, "%s\\Sample.ini", cur_dir);
    GetPrivateProfileString("GENERAL",

                                                                    "LD",
                                                                    (LPCTSTR)szRet,
                                                                    128,
                                                                    szIniFile);

    if (!strcmp("YES", LPSTR, szRet))
        GetPrivateProfileString("SETTINGS",

                                                                    (LPCTSTR)pszKey,
                                                                    "",
                                                                    (LPCTSTR)szRet,
                                                                    128,
                                                                    szIniFile);

    if (!strcmp((LPCTSTR)szRet, ""))
        return TRUE;
    return TRUE;
}

LoadString(AfxGetInstanceHandle(), nID, szRet, 128);
wsprintf(LPSTR, pszRet, (LPCTSTR)szRet);
return TRUE;
}
//Load String

```

【図5】

```
[GENERAL]
LD YES

[STRINGS]
IDS_STR_TITLE="デバイス構成"
IDS_STR_SERIAL="シリアル・ポート"
IDS_STR_PARA="パラレル・ポート"
IDS_STR_MODEM="モデム"
IDS_STR_IR="赤外線ポート"
IDS_STR_AUDIO="オーディオ"
IDS_STR_EN="使用する"
IDS_STR_DIS="使用しない"
IDS_STR_OK="了解(O)"
IDS_STR_HELP="ヘルプ(&H)"
```

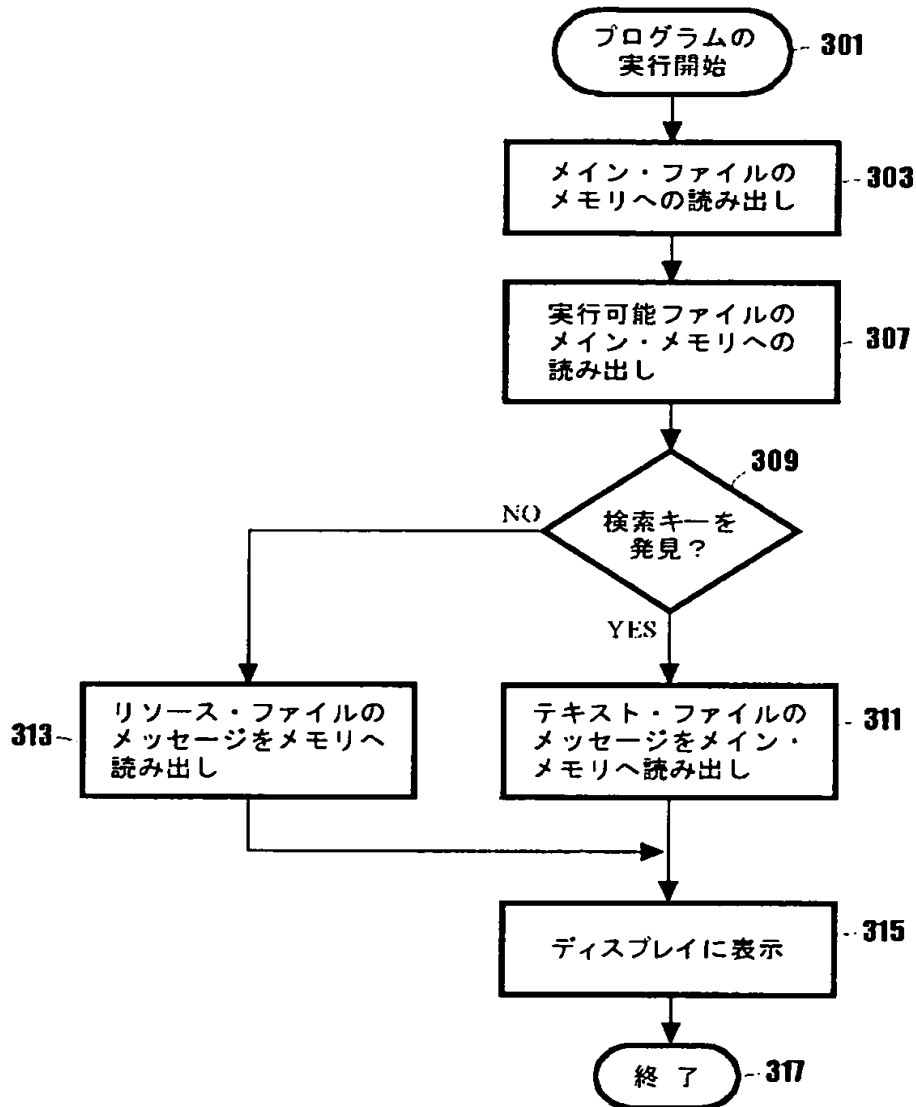
(A)

```
[GENERAL]
LD YES

[STRINGS]
IDS_STR_TITLE="Configuration de périphérique"
IDS_STR_SERIAL="Port série"
IDS_STR_PARA="Port parallèle"
IDS_STR_MODEM="Modem interne"
IDS_STR_IR="Port infrarouge"
IDS_STR_AUDIO="Audio"
IDS_STR_EN="Activer"
IDS_STR_DIS="Désactiver"
IDS_STR_OK="OK"
IDS_STR_HELP="Aide"
```

(B)

【図 6】



【書類名】 要約書

【要約】

【課題】 プログラムの実行に伴うGUIに表示するキャラクタ・ストリングを他国語に翻訳して表示する方法を提供する。

【解決手段】 コンピュータ・プログラムが提供するGUI環境にキャラクタ・ストリングを表示する方法であって、（a）実行可能プログラムを提供するステップと、（b）前記キャラクタ・ストリングを含み前記実行可能プログラムから参照することが可能なテキスト・ファイルを提供するステップと、（c）前記実行可能プログラムを実行するステップ（301）と（d）前記実行可能プログラムから前記テキスト・ファイルを参照するステップ（307）と、（e）前記参照したテキスト・ファイルに含まれるキャラクタ・ストリングを表示するステップ（315）とを有する。

【選択図】 図6

認定・付加情報

特許出願の番号	特願2000-034610
受付番号	50000158294
書類名	特許願
担当官	第七担当上席 0096
作成日	平成12年 2月15日

<認定情報・付加情報>

【提出日】	平成12年 2月14日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 1990年10月24日
[変更理由] 新規登録
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション